

Morfologia matemática binária:

- Afinamento

Morfologia matemática cinza:

- Dilatação
- Erosão
- Gradiente por erosão e Gradiente por dilatação

- erosão de A por B: $A \ominus B = \{z | (B)_z \subseteq A\}$

- dilatação de A por B: $A \oplus B = \{z | (B)_z \cap A \neq \emptyset\}$

- abertura de A por B: $A \circ B = (A \ominus B) \oplus B.$

- fechamento de A por B: $A \bullet B = (A \oplus B) \ominus B.$

Acerto e Erro (*Hit-or-miss*): $A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$

Afinamento

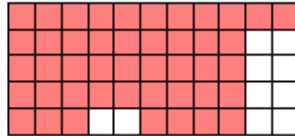
Serra (1982): Afinamento de um conjunto A por um elemento estruturante B:

$$A \circledast B = A - (A \otimes B)$$

onde

$$A \otimes B = (A \ominus B_1) \cap (A \ominus B_2) \quad \text{é o operador } \textit{hit or miss} \text{ (acerto e erro)}.$$

A é a imagem :



Morfologia matemática em nível de cinza com elemento estruturante plano:

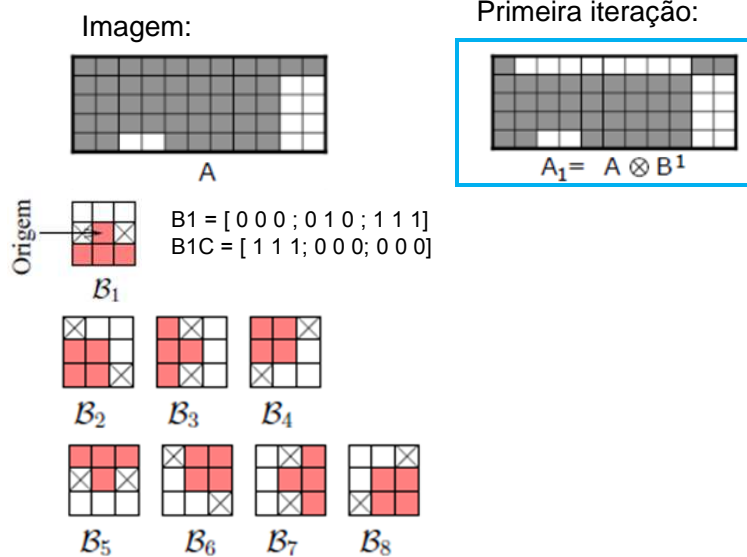
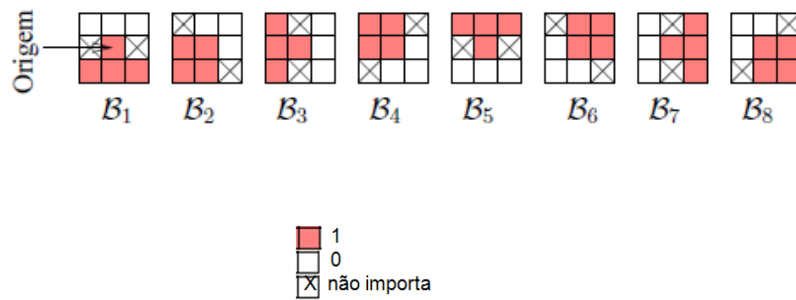
- Erosão
- Dilatação

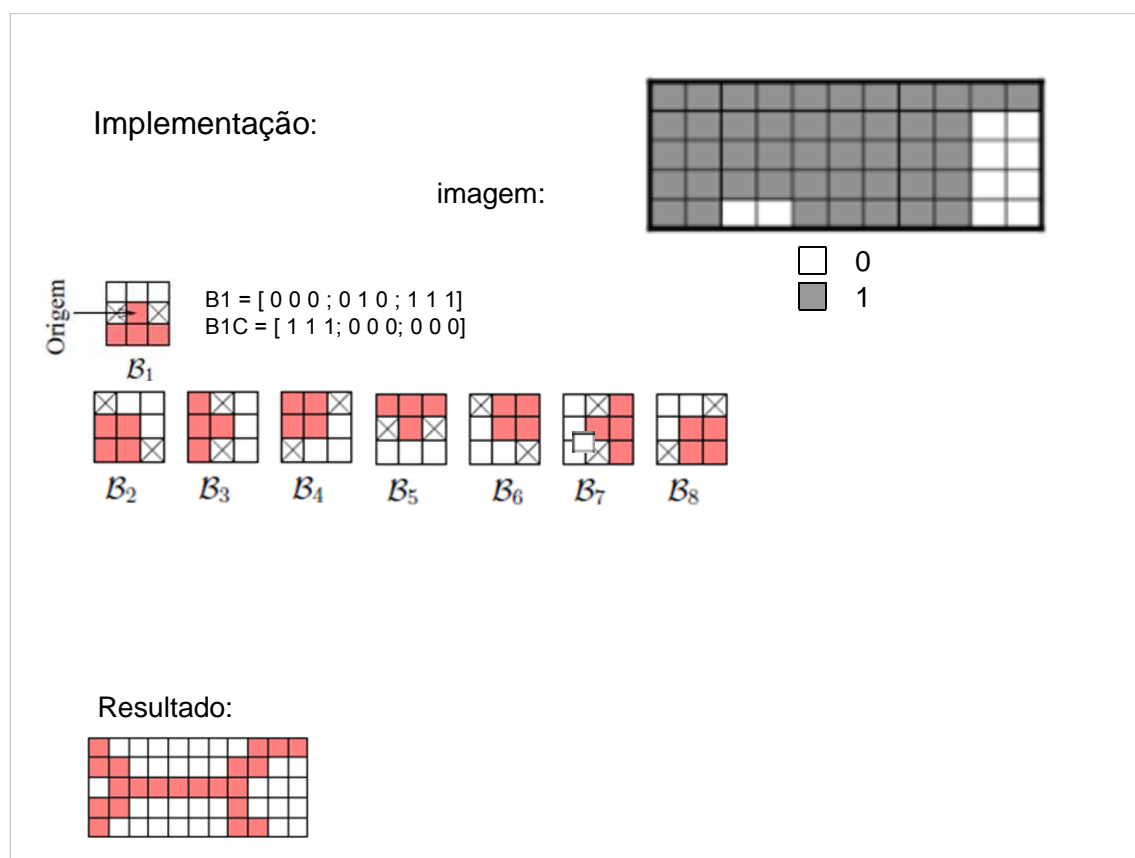
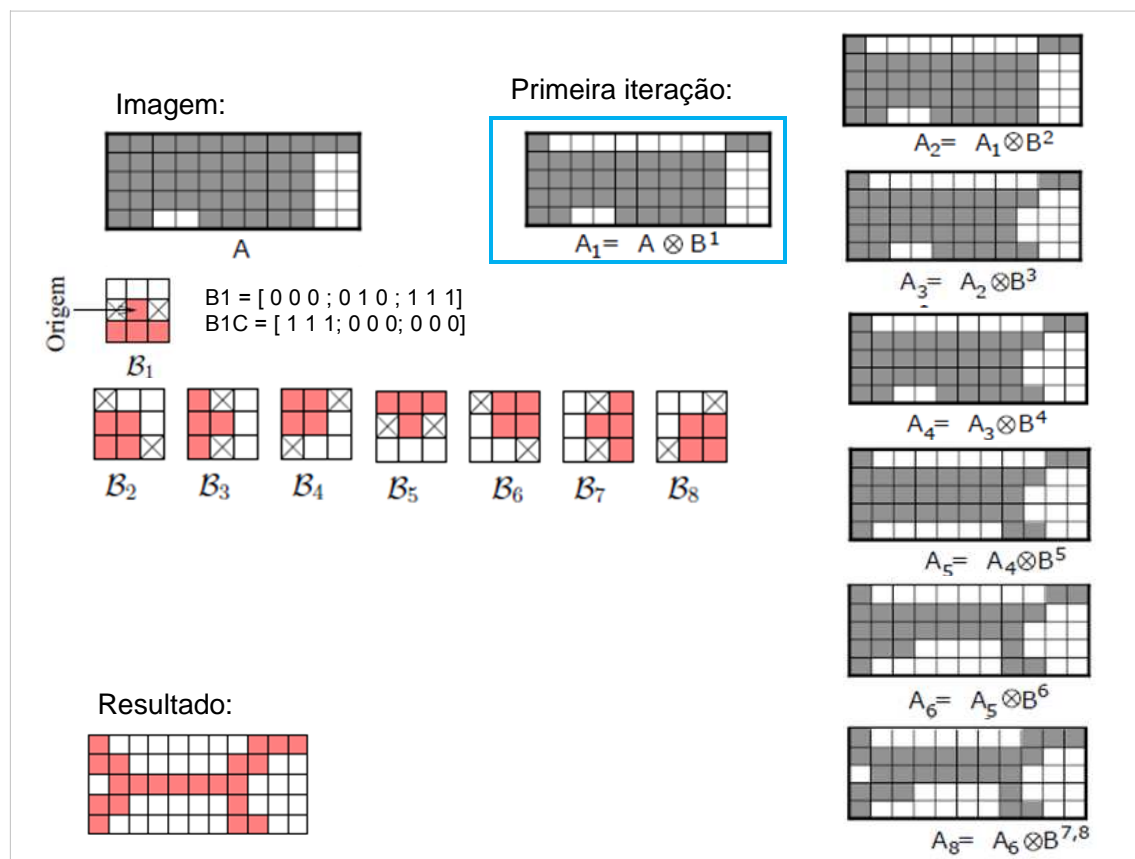
Afinamento

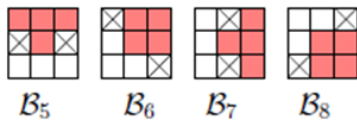
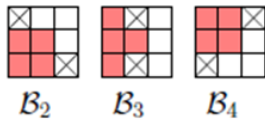
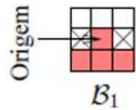
Outra implementação do afinamento de um conjunto A pelo elemento estruturante B:

$$A \circledast \{B\} = (((((A \otimes B_1) \otimes B_2) \otimes \dots) \otimes B_n)$$

Utiliza-se um conjunto B de elementos estruturantes $B_i = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8\}$







%Elementos estruturantes

$B_1 = [0\ 0\ 0; 0\ 1\ 0; 1\ 1\ 1]$

$B_1C = [1\ 1\ 1; 0\ 0\ 0; 0\ 0\ 0]$

$B_2 = [0\ 0\ 0; 1\ 1\ 0; 1\ 1\ 0]$

$B_2C = [0\ 1\ 1; 0\ 0\ 1; 0\ 0\ 0]$

$B_3 = [1\ 0\ 0; 1\ 1\ 0; 1\ 0\ 0]$

$B_3C = [0\ 0\ 1; 0\ 0\ 1; 0\ 0\ 1]$

$B_4 = [1\ 1\ 0; 1\ 1\ 0; 0\ 0\ 0]$

$B_4C = [0\ 0\ 0; 0\ 0\ 1; 0\ 1\ 1]$

$B_5 = [1\ 1\ 1; 0\ 1\ 0; 0\ 0\ 0]$

$B_5C = [0\ 0\ 0; 0\ 0\ 0; 1\ 1\ 1]$

$B_6 = [0\ 1\ 1; 0\ 1\ 1; 0\ 0\ 0]$

$B_6C = [0\ 0\ 0; 1\ 0\ 0; 1\ 1\ 0]$

$B_7 = [0\ 0\ 1; 0\ 1\ 1; 0\ 0\ 1]$

$B_7C = [1\ 0\ 0; 1\ 0\ 0; 1\ 0\ 0]$

$B_8 = [0\ 0\ 0; 0\ 1\ 1; 0\ 1\ 1]$

$B_8C = [1\ 1\ 0; 1\ 0\ 0; 0\ 0\ 0]$

```
[m,n] = size(I);
I1 = zeros(m,n,8);
S1 = zeros(m,n);
S2 = zeros(m,n);
S = zeros(m,n);
```

```
it = 1
```

```
Filtro = 1
```

```
while it < 10
```

```
while Filtro < 9
```

```
% escolher filtros Bi e BiC
```

```
% Bi = ...
```

```
% BiC = ...
```

```
% Aplicar filtros
```

```
laux = I - S;
```

```
I1(:, :, Filtro) = laux;
```

```
I = laux;
```

```
Filtro = Filtro + 1;
```

```
end
```

```
it = it + 1;
```

```
Filtro = 1;
```

```
end
```

```
if Filtro == 1
```

```
Bi = B1;
```

```
BiC = B1C;
```

```
elseif Filtro == 2
```

```
Bi = B2;
```

```
BiC = B2C;
```

```
elseif Filtro == 3
```

```
Bi = B3;
```

```
BiC = B3C;
```

```
elseif Filtro == 4
```

```
Bi = B4;
```

```
BiC = B4C;
```

```
elseif Filtro == 5
```

```
Bi = B5;
```

```
BiC = B5C;
```

```
elseif Filtro == 6
```

```
Bi = B6;
```

```
BiC = B6C;
```

```
elseif Filtro == 7
```

```
Bi = B7;
```

```
BiC = B7C;
```

```
elseif Filtro == 8
```

```
Bi = B8;
```

```
BiC = B8C;
```

```
end
```

```

m,n] = size(I);
I1 = zeros(m,n,8);
S1 = zeros(m,n);
S2 = zeros(m,n);
S = zeros(m,n);
it = 1
Filtro = 1
while it < 18

while Filtro < 9
    % escolher filtros Bi e BiC
    % Bi = ...
    % BiC = ...

    % Aplicar filtros

    laux = I - S;
    I1(:, :, Filtro) = laux;
    I = laux;

    Filtro = Filtro + 1;
end

it = it + 1;
Filtro = 1;

end

```

```

for i = 2:m-1
for j = 2:n-1

    ind = 0;
    for k = -1:1
    for l = -1:1
        if Bi2+k, 2+l) == 1 & I(i+k,j+l) == 1
            ind = ind+1;
        end
    end
    end
    if ind == 4
        S1(i,j) = I(i,j);
    else
        S1(i,j) = 0;
    end

    ind = 0;
    for k = -1:1
    for l = -1:1
        if BiC(2+k, 2+l) == 1 & (1 - I(i+k,j+l)) == 1
            ind = ind+1;
        end
    end
    end
    if ind >= 3
        S2(i,j) = 0; % 1-I(i+k,j+l) ;
    else
        S2(i,j) = 1;
    end

end
end
for i = 2:m-1
for j = 2:n-1
    if S1(i,j) == 1 & S2(i,j) == 0
        S(i,j) = 1;
    else
        S(i,j) = 0;
    end
end
end
end
end

```

Exercício 1:

Aplice o operador afinamento à imagem 'ferramenta5.tif' considerando 4 iterações:



Exercício 4:

Como tornar a implementação do operador afinamento independente do número de iterações?

Aplique o programa modificado à imagem *ferramenta5.tif* e verifique quantas iterações foram necessárias para obter o esqueleto da figura.



```
m,n] = size(I);
I1 = zeros(m,n,8);
S1 = zeros(m,n);
S2 = zeros(m,n);
S = zeros(m,n);
it = 1
Filtro = 1
Max = 10
while Max > 0
    while Filtro < 9
        % escolher filtros B11 e B11C
        % Bi = ...
        % BiC = ...

        % Aplicar filtros

        laux = I - S;
        I1(:, :, Filtro) = laux;
        I = laux;
        Filtro = Filtro + 1;
    end

    % erosão da imagem I
    % parar quando Max == 0

    it = it+1;
    Filtro = 1;
end
% imagem de saída: I

% para EE quadrado
EE = [ 1 1 1; 1 1 1; 1 1 1];
S3 = zeros(m,n);

for i = d1:m-d2
    for j = d1:n-d2
        ind = 0;
        for k = -d2:d2
            for l = -d2:d2
                if EE(d1+k,d1+l) == 1 & I(i+k,j+l) == 1
                    ind = ind+1;
                end
            end
        end
        if ind == 9
            S3(i,j) = 1;
        end
    end
end

Max = max(max(S3));
```

Exercício 1-A:

Como tornar a implementação do operador afinamento independente do número de iterações?

Aplique o programa modificado à imagem *ferramenta5.tif* e verifique quantas iterações foram necessárias para obter o esqueleto da figura.



```
m,n] = size(I);
I1 = zeros(m,n,8);
S1 = zeros(m,n);
S2 = zeros(m,n);
S = zeros(m,n);
it = 1
Filtro = 1
Max = 10
while Max > 0
    while Filtro < 9
        % escolher filtros B11 e B11C
        % Bi = ...
        % BiC = ...

        % Aplicar filtros

        laux = I - S;
        I1(:, :, Filtro) = laux;
        I = laux;
        Filtro = Filtro + 1;
    end

    % erosão da imagem I
    % parar quando Max == 0

    it = it+1;
    Filtro = 1;
end
% imagem de saída: I

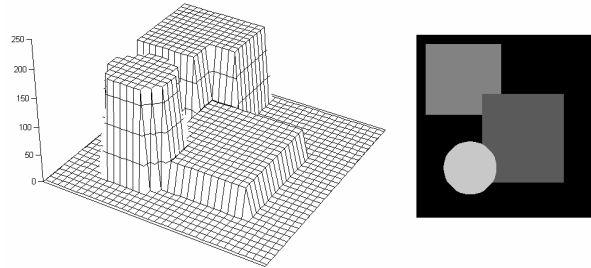
% para EE quadrado
EE = [ 1 1 1; 1 1 1; 1 1 1];
S3 = zeros(m,n);

for i = d1:m-d2
    for j = d1:n-d2
        ind = 0;
        for k = -d2:d2
            for l = -d2:d2
                if EE(d1+k,d1+l) == 1 & I(i+k,j+l) == 1
                    ind = ind+1;
                end
            end
        end
        if ind == 9
            S3(i,j) = 1;
        end
    end
end

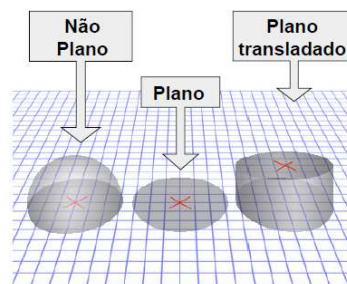
Max = max(max(S3));
```


Morfologia em nível de cinza

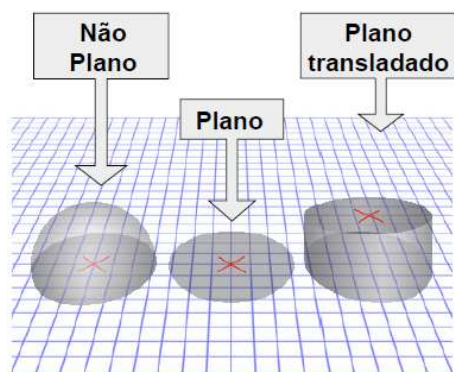
-Uma imagem em nível de cinza pode ser vista como uma objeto 3D , em que a 3ª dimensão é o nível de cinza



-Elemento estruturante



- Elemento estruturante:



Dilatação e erosão com elemento estruturante não planar

Elementos estruturantes:

O	1	0
1	2	1
0	1	0

1	2	1
2	3	2
1	2	1

- A dilatação da imagem A pelo elemento estruturante B é expresso por:

$$(A \oplus B)(i,j) = \max\{A(i-k, j-l) + B(k,l)\}$$

20	23	26
17	19	19
34	36	27
32	27	18
34	27	25

28	32	25	25	17
35	28	34	33	28
33	37	44	40	41
16	21	26	28	32
23	24	35	37	29

1	2	1
2	3	2
1	2	1

-	-	-	-	-	-	-	-
-	38						-
-							-
-							-
-	-	-	-	-	-	-	-

- A erosão da imagem A pelo elemento estruturante B é expresso por:

$$(A \ominus B)(i,j) = \min\{A(i-k, j-l) - B(k,l)\}$$

20	23	26
17	19	19
34	36	27
32	27	18
34	27	25

28	32	25	25	17
35	28	34	33	28
33	37	44	40	41
16	21	26	28	32
23	24	35	37	29

1	2	1
2	3	2
1	2	1

-	-	-	-	-	-	-	-
-	15						-
-							-
-							-
-	-	-	-	-	-	-	-

```

I = [
20 23 26 28 32 25 25 17
17 19 19 35 28 34 33 28
34 36 27 33 37 44 40 41
32 27 18 16 21 26 28 32
34 27 25 23 24 35 37 29
]

EE = [
1 2 1
2 3 2
1 2 1
]

% dilatação
[m,n] = size(I);
[me,ne] = size(EE);

d2 = ceil(me/2);
d1 = floor(me/2);

S1 = I;

for i = d2:m-d1
    for j = d2:n-d1
        a = zeros(1,mf*mf);

        ind = 1;
        for k = -d1:d1
            for l = -d1:d1
                a(1,ind) = I(i+k, j+l) + EE(k,l);
                ind = ind+1;
            end
        end

        S1(i,j) = max(a);
    end
end

```

```

I = [
20 23 26 28 32 25 25 17
17 19 19 35 28 34 33 28
34 36 27 33 37 44 40 41
32 27 18 16 21 26 28 32
34 27 25 23 24 35 37 29
]

EE = [
1 2 1
2 3 2
1 2 1
]

% erosão
[m,n] = size(I);
[me,ne] = size(EE);

d2 = ceil(me/2);
d1 = floor(me/2);

S1 = I;

for i = d2:m-d1
for j = d2:n-d1
    a = zeros(1,me*me);

    ind = 1;
    for k = -d1:d1
    for l = -d1:d1
        a(ind) = I(i+k, j+l) - EE(d2+k,d2+l);
        ind = ind+1;
    end
    end

    S1(i,j) = min(a);

end
end

```

Dilatação: $Dil(I) = \max\{I(i-k, j-k)\}$

Elemento estruturante = ?

```

for i = d2:m-d1
for j = d2:n-d1
    a = zeros(1,mf*mf);

    ind = 1;
    for k = -d1:d1
    for l = -d1:d1
        a(ind) = I(i+k, j+l);
        ind = ind+1;
    end
    end

```

S1(i,j) = max(a);

```

end
end

```

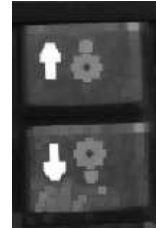
Consiste em substituir o ND (i,j) original pelo máximo ND correspondente à atual posição do elemento estruturante.

Efeito: alargamento das regiões claras.

Imagem I



Dil(I)



Erosão: $Ero(I) = \min\{I(i-k, j-k)\}$

```

for i = d2:m-d1
for j = d2:n-d1
    a = zeros(1,me*me);

    ind = 1;
    for k = -d1:d1
        for l = -d1:d1
            a(ind) = I(i+k, j+l);
            ind = ind+1;
        end
    end

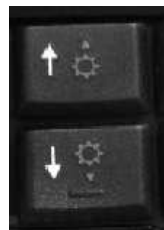
    S1(i,j) = min(a);
end
end

```

Consiste em substituir o ND (i,j) original pelo mínimo ND correspondente à atual posição do elemento estruturante.

Efeito: alargamento das regiões escuras.

Imagem I



Ero(I)



Abertura e fechamento

-A abertura consiste de uma erosão de tons de cinza seguida de uma dilatação de tons de cinza usando o mesmo elemento estruturante.

-O fechamento consiste de uma dilatação de tons de cinza seguida de uma erosão usando o mesmo elemento estruturante.

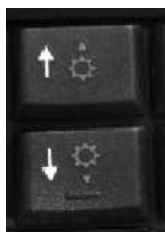
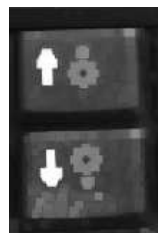


Imagem I

Dil(I)

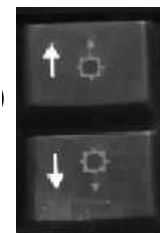


Ero(I)



Fechamento:

$$I \bullet B = (I \oplus B) \ominus B.$$



Abertura:

$$I \circ B = (I \ominus B) \oplus B.$$



- Exercício 2: Retirar pequenos detalhes, deixando a imagem mais homogênea.

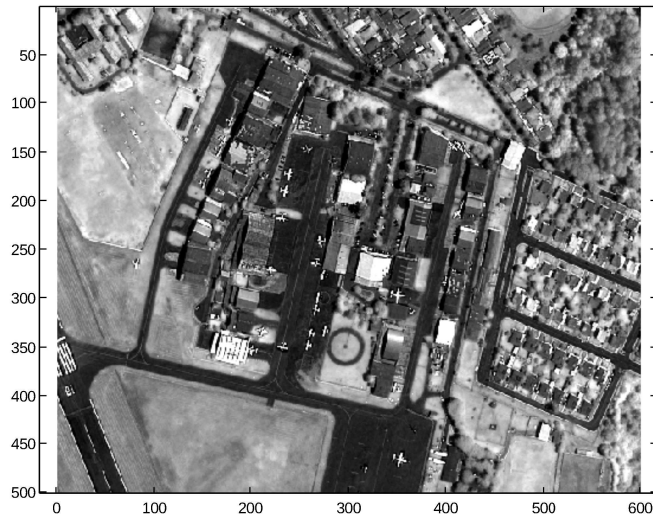
-> detalhes claros: aviões

Imagem: Bacacheri.tif

Análise:

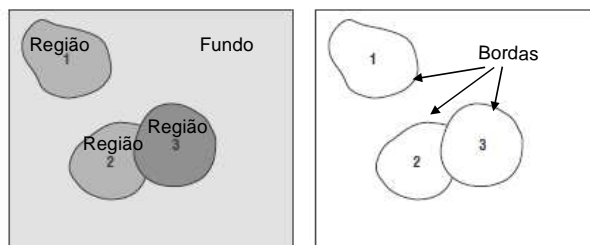
-Qual operador morfológico deve ser utilizado para obter os efeitos desejados?

-Qual deve ser o tamanho do elemento estruturante?



Derivada e gradiente

- > Implementação de derivadas em filtros que ressaltam bordas e transições .
- Bordas em imagens digitais frequentemente se comportam como transições de intensidade.



- Detecção de bordas: localizar os pixels das bordas;
- Realce de bordas : aumentar o contraste entre as bordas e o fundo.

Derivada e gradiente

- Bordas **verticais** podem ser detectadas pela **diferença horizontal** entre pontos adjacentes,
- Bordas **horizontais** podem ser detectadas pela **diferença vertical** entre pontos adjacentes.

O gradiente de uma função de duas variáveis $f(x,y)$, nas coordenadas (x,y) é definido por:

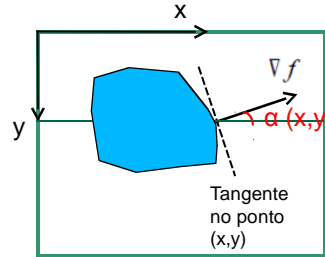
$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

a magnitude deste vetor é:

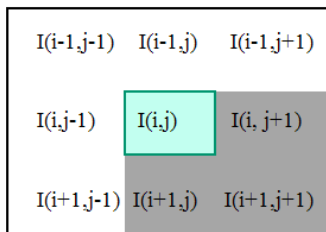
$$|\nabla f| = [G_x^2 + G_y^2]^{1/2}$$

A direção do vetor gradiente também é uma característica importante para muitas aplicações, ela é dada por:

$$\alpha(x,y) = \arctan \frac{G_y}{G_x}$$



Implementações do cálculo de gradiente:



Considerando uma máscara percorrendo a imagem, sendo $I(i,j)$ a posição atual sobre a imagem.

- Operador Roberts (1965), considera a vizinhança em cinza ,
- Operador Prewitt (1966), considera a região 3x3
- Operador Sobel (1970), considera a região 3x3

Roberts:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Prewitt:

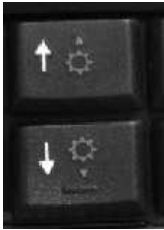
$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Sobel:

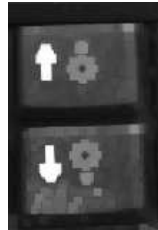
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Gradiente morfológico:

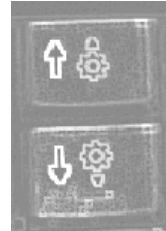
Imagem I



Dil(I)



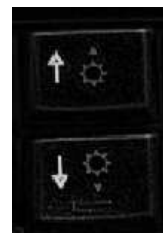
Gradiente dilatado = Dil(I) - I



Ero(I)



Gradiente erodido = I - Ero(I)



- Exercício 3: Extrair os contornos aplicando

a) Gradiente por dilatação

b) Gradiente por erosão

Imagem: teclado2.tif

Analisar:

-Qual operador morfológico deve ser utilizado para obter os efeitos desejados?

-Qual deve ser o tamanho do elemento estruturante?

